Biological signalling and causality

Vincent Danos, Walter Fontana, Russ Harmer, Jean Krivine

Biological signalling and causality

Vincent Danos* Walter Fontana† Russell Harmer‡ Jean Krivine §

May 17, 2007

Abstract

Modelling is becoming a necessity in studying biological signalling pathways, because the combinatorial complexity of such systems rapidly overwhelms intuitive and qualitative forms of reasoning. Yet, this same combinatorial explosion makes the traditional modelling paradigm based on systems of differential equations quickly impractical, if not conceptually inappropriate. As an alternative, we propose an agent-based / concurrent language, named κ [4–8], which places causal reasoning front center. We illustrate how κ transparently represents biological knowledge, thereby making models easier to build, discuss, modify, and merge. By taming the combinatorial explosion, circumventing the frustrations of handling opaque systems of equations, and lowering the mathematical threshold for molecular biologists, κ holds promise for making modelling more widely available.

The causal structure of processes [9], is largely absent from systems of differential equations, yet it deeply shapes the dynamical, and perhaps even evolutionary, characteristics of complex distributed biological systems. We illustrate the use of κ and its associated causal analysis by means of a model of EGFR signalling that would overwhelm any traditional approach. The model is obtained by refactoring two extant models based on differential equations [3, 19]. We formalize the colloquial concept of pathway in terms of a special kind of event structure and illustrate how the juxtaposition of it with relationships of conflict between rules can be used to dissect EGFR signalling dynamics.

1 Background

A large majority of models aimed at investigating the behavior of biological pathways is cast in terms of systems of differential equations [3,12,14,15,17,19]. The choice seems natural. The theory of dynamical systems offers an extensive repertoire of mathematical techniques for reasoning about such networks. It provides, at least in the limit of long times, a well-understood ontology of behaviors, like steady states, oscillations, and chaos, along with their (linear) stability properties. The ready availability of numerical procedures for integrating systems of equations, while varying over parameters and initial conditions, completes a powerful workbench that has successfully carried much of physics and chemical kinetics. Yet, this workbench is showing clear signs of cracking under the ponderous combinatorial complexity of molecular signalling processes, which involve proteins

*Plectix Biosystems
†Harvard Medical School
‡CNRS, Université Denis Diderot
§École Polytechnique

that interact through multiple post-translational modifications and physical associations within an intricate topology of locales [18].

Representations of chemical reaction networks in terms of differential equations are about chemical kinetics, not the unfolding of chemistry. In fact, all molecular species made possible by a set of chemical transformations must be explicitly known in advance for setting up the corresponding system of kinetic equations. Every molecular species has its own concentration variable and an equation describing its rate of change as imparted by all reactions that produce or consume that species. These reactions, too, must be known in advance. Many ion channels, kinases, phosphatases, and receptors – to mention a few – are proteins that possess multiple sites at which they can be modified by phosphorylation, ubiquitination, methylation, glycosidilation, and a plethora of other chemical tagging processes. About one in a hundred proteins have at least 8 modifiable sites, which means 256 states. A simple heterodimer of two distinct proteins, each with that much state, would weigh in at more than 65,000 equations. It is easily seen that this combinatorics can rapidly amount to more possible chemical species than can be realized by the actual number of molecules involved in a cellular process of this kind. The problem is not so much that a deterministic description is no longer warranted, but rather that the equations, whether deterministic or stochastic, cannot be written down anymore. And if they could, what would one learn from them?

This difficulty is well recognized. One way out is to use aggregate variables decribing sets of modification forms. For example, one might bundle together all phosphoforms of a receptor, regardless of which sites are phosphorylated. This, however, is problematic. First, the choice of what to aggregate and not is unprincipled. Second, the appropriate level of aggregation may change over time as the system dynamics unfolds. Third, the aggregation is error prone, since it has to be done without a prior microscopic description. A further, more subtle, difficulty is that an extensional system of differential equations describes the constituent molecules only in terms of interactions that are relevant in a given context of other molecules. It does not characterize molecular components in terms of their potential interactions that could become relevant if the composition of the system were to change. As a consequence, "compositional perturbations", such as adding a novel molecular component (a drug) or modifying an extant one (to model the effects of knocking out a site or adding a new domain) are virtually impossible to carry out by hand, since they require, again, enumerating all chemical consequences in advance and then rewriting all affected equations.

These problems have led to recent attempts at describing molecular reaction networks in terms of molecules as "agents", whose possible interactions are defined by rules that specify how a local pattern of "sites" and their "states" is to be rewritten [2, 10]. This resembles good old organic chemistry, except that biologists think of post-translational modifications less as chemical transformations (which, of course, they ultimately are) than as state changes of the *same* agent. A phosphorylated kinase is, at some useful level of description, still the same entity - though in a different state - than its unphosphorylated version. Indeed, biologists think of the state of an agent as a specific set of interaction capabilities. The discontinuous change of such capabilities despite an underlying continuity in agent-type hinges on the large size of a protein, which allows for a significant change in hydrophobic and electrostatic dispositions without much changing the protein's overall chemical identity. In contrast, chemists dont think of oxaloacetate as being a different state of pyruvate.

A rule may specify, for example, that if site Y996 of protein A is phosphorylated, protein B can bind to it with its site SH2. Since this rule applies regardless of whether A or B are bound to other partners or possess other sites with particular states, it captures a potentially large set of

individual reactions between distinct molecular species. The need for spelling out all these reactions was spelling problems for "flat" (extensional) reaction network representations, whereas modifying or extending a reaction system is now as simple as modifying a single rule or merging sets of rules, respectively.

Our stance in this paper is to forgo writing out differential equations. Instead, we directly operate at the level of rules defining the interactions among a set of agents. The principal challenge is to do so without abandoning the possibility for formal analysis. In the dynamical systems framework, the set of differential equations was a formal object amenable to analysis - even if that analysis involves approximations (such as the adiabatic elimination of variables), which are still rigorous procedures. When replacing a set of differential equations with a set of rules that rewrite patterns defined on agents and combinations of agents, we should define formal procedures with which to extract statements about the possible behaviors of a system governed by that set of rules. In this paper we provide a starting point, mainly in form of an illustrative example, that will be developed more rigorously elsewhere.

The stance we take is strongly influenced by how computer scientists reason about concurrent or distributed systems, and in particular by the causal analysis of distributed systems [1,9]. Biological signalling and control processes are, in fact, massively distributed systems. Taking concurrency seriously means understanding the organization of such systems in terms of observables defined from within rather than outside these systems. Time is a particular case in point. In molecular systems, the temporal precedence among events cannot be defined (at first) on physical time, since cells or molecules do not bear watches, let alone synchronized ones. It is well known in concurrency that temporal precedence is a logical relation that gives rise to a partial order, as opposed to a total order. Some events must occur before others can happen, while other events may happen in any sequence, reflecting their mutual independence. Clearly, in any particular physical realization one will observe a particular sequence of events. The issue, however, is to uncover which aspects of that sequence are necessary and which contingent. The issue is to discover the invariant structure underlying all observable sequences. That structure is derived from the concept of event structure [21] and represents the form of causality we seek to illustrate in a biological example. Differential equations are unable to resolve this causality, precisely because they treat time as global, as if everything proceeded in a synchronized fashion. In contrast, the rule-based approach seeks to understand the dependencies that constrain an observable event, such as the production of a particular molecular complex. The rule-based approach permits to clarify the causal relationships between rules that explain how a path toward a specified event has unfolded from initial conditions. Such a path seems an appropriate formalization of what biologists colloquially call a "signalling pathway".

There are many paths towards a particular event. It therefore becomes paramount to isolate those paths that occur with significant frequency from a given starting condition. In addition, paths are themselves time-dependent in reflection of the changing composition of resources available to their occurrence. To sample pathways, requires fast and scalable stochastic simulation tools, which we have implemented and described elsewhere [5].

Last, but not least, a consistent rule-based framework permits the storage and clean update of all knowledge about the interaction capabilities of a biomolecular agent, regardless of whether all of these capabilities are needed in a particular model of signalling. A rule need only mention the interface of an agent that is relevant to that rule. In this sense, rules and agents expressed in a formal grammar represent empirical knowledge that is executable.

Here we sketch some early steps towards developing formal analytical tools that, in conjunction with stochastic simulation, can generate insights into the collective properties of distributed rule-

based systems. In outline, we build a model that would be quite large and unwieldy by traditional standards, but appears fairly simple within our framework. We then construct causal histories, called *stories*, of the type described above. On the basis of relationships of activation and inhibition (or conflict) between rules, we identify key junctures in the system's dynamics that yield explanatory insights and suggest numerical experiments.

To introduce the framework, we warm up with a simple example of an ubiquitous control motif in cellular signal processing: a futile cycle of enzymatic modification and demodification of a target substrate.

2 A futile cycle

2.1 Agents and rules

The κ description of a system consists of a collection of *agents* and *rules*. An agent has a name and a number of labeled sites, collectively referred to as the agent's interface. A site may have an internal state, typically used to denote its phosphorylation status or other post-translational modification. Rules provide a concise description of how instances of agents interact. Elementary interactions consist of the binding or unbinding of two agents and the modification of the state of a site. This seems limited, but closely matches the style of reasoning that molecular biologists apply to mechanistic interactions in cellular signalling. While this approach does not address all aspects of signaling (such as compartmentation), it does cover a substantive body of events sufficient for the present purpose.

To develop the main concepts, we start with a system consisting of three agents: a kinase K, a target T with two phosphorylatable sites, and a phosphatase P. We first describe a phosphorylation event by means of three elementary actions and their corresponding rules: (1) the kinase K binds its target T either at site x or y; (2) the kinase may (but need not) phosphorylate the site to which it is bound; (3) the kinase dissociates (unbinds) from its target. For ease of reference, we label rules with a mnemonic on the left. Using a textual notation, we represent internal states as '`~u`' (unphosphorylated), and '`~p`' (phosphorylated), and physical associations (bindings or links) as '`!`' with shared indices across agents to indicate the two endpoints of a link. The left hand side of a rule specifies a condition in the form of a pattern expressed as a partial graph, which represents binding states and site values of agents. The right hand side of a rule specifies (usually elementary) changes to agents mentioned on the left. A double arrow indicates a reversible rule. With these conventions, the phosphorylation process of sites x or y translates into:

```
'KT@x' K(a),T(x) <-> K(a!1),T(x!1)
'Tp@x' K(a!1),T(x~u!1) -> K(a!1),T(x~p!1)
'KT@y' K(a),T(y) <-> K(a!1),T(y!1)
'Tp@y' K(a!1),T(y~u!1) -> K(a!1),T(y~p!1)
```

Likewise, the action of the phosphatase P, which undoes the action of K, is described by a set of dual rules:

```
'PT@x' P(a),T(x) <-> P(a!1),T(x!1)
'Tu@x' P(a!1),T(x~p!1) -> P(a!1),T(x~u!1)
'PT@y' P(a),T(y) <-> P(a!1),T(y!1)
'Tu@y' P(a!1),T(y~p!1) -> P(a!1),T(y~u!1)
```

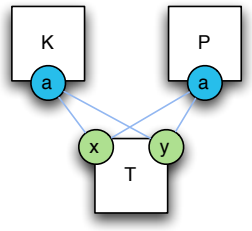


Figure 1: *A Goldbeter-Koshland contact map: nodes represent the three kinds of agents in the rule set, together with their sites, and each site is connected sites it can bind to; whether a site can be modified is indicated by a colour code (green). Although very simple, the associated rule set generates already 38 non-isomorphic complexes (36 of which contain T).*

It is possible to associate rate constants with each rule, as we shall do later. We refer to this rule set as the Goldbeter-Koshland (GK) loop [11]. It is a frequent motif that appears in many variants throughout cellular signal transduction. Notice how the specification of elementary actions forces us to make our mechanistic choices explicit. The example views phosphorylation of T by K as a distributed mechanism, whereby the kinase lets go of its target before phosphorylating it (or another instance) again, since it cannot remain bound to site x and phosphorylate site y. Other variants of multisite phosphorylation involve a processive mechanism whereby the same kinase acts sequentially on some or all sites of its target. Further variants still would have to specify whether multiple kinases can be bound to the same target or not.

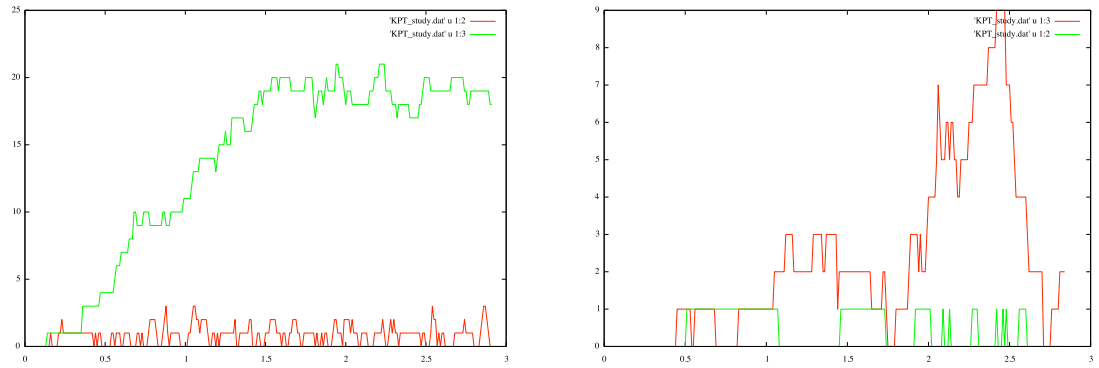
2.2 The contact map

Large rule sets can be difficult to understand, although they are nowhere nearly as opaque as models based on thousands of differential equations. It is therefore important to have a suite of views that report at a glance information implied by the rules. Such cognitive devices are useful for a piecewise modular construction of the system of interest. They also allow for a modicum of reasoning about the system. A first useful view of the rule set is the *contact map*, which is akin to a protein-protein interaction (PPI) map and is shown in Fig. 1. The contact map is a graph whose nodes are the agents with their interfaces and whose edges represent possible bindings between sites. Potential site modifications are indicated by a colour code. The contact map does not provide causal information, in the sense that it does not specify which conditions must be met for a modification or binding to occur. It only shows possibilities. It may be thought of as a static typing of a rule set.

2.3 Trajectories

With a rule set in place, one can generate time courses, or stochastic trajectories, for user-defined *observables*. One selects an initial state, which here consists of 100 instances of each of the three agents with their interfaces in defined states, `T(x~u,y~u)`, `K(a)`, `P(a)`. In addition, rate constants may be specified for each rule (see the caption to Fig. 2(a)). We chose to track two observables: (1) the number of doubly phosphorylated target molecules, regardless of whether sites x and y are bound to enzymes, which is written as `T(x~p?,y~p?)`, and (2) the number of target instances T

that are fully phosphorylated but free on both sites x and y, `T(x~p,y~p)`. As can be verified in Fig. 2(a), the latter observable has to be smaller since it is more stringent. The trajectories are obtained using an entirely rule-based continuous time Markov chain (Gillespie) kinetics [5]. At any given time a rule can apply to the set of instances in the system in a number of ways. That number is multiplied by the rate of the rule and defines the rule's *activity* or flux. It determines the likelihood that this rule will fire next. The total activity of the system determines probabilistically the associated (simulated) time advance. One has to keep in mind that such trajectories are but one realization of a stochastic process that will differ slightly when repeated. Statistical properties require averaging over many runs, but we shall not be concerned with them here.



(a) *Association and modification rates are set to 1, and dissociation rates are set to 10 (per time units).*

(b) *Same model perturbed by a tenfold increase in the association and modification rate of P at site x.*

Figure 2: *Simulation of the Goldbeter-Koshland loop: the initial state has a hundred instances of each agent, each disconnected and unphosphorylated; the level of doubly phosphorylated target is lower in the perturbed case (right).*

2.4 Causality and stories

The contact map and the trajectory samples represent an *agent-centric* view of the system's evolution. They do not answer directly the question of which succession of events results in a fully phosphorylated form of the target T. This is where the notion of pathway or *story* comes into play. An event is said to cause another event, if the former cannot possibly happen after the latter. An example is an event of type `Tu@x`, which requires the prior occurrence of an event of type `PT@x`. This notion of causation defines a partial order on any sequence of events. The fact that two events may succeed one another in a particular trajectory does not imply that they are in a causal relationship. An example is provided by two successive events of type `KT@x` and `KT@y`. Events that are not related by precedence are said to be concurrent.

By a story we mean the causal lineage that led to an instance of a user-defined observable. Causality is an acyclic relationship between events (instances of rule applications), not agents. An observable in a story must therefore be a rule that produces a particular molecular configuration that one wishes to observe. If there are many rules, one may define an "identity-rule", which (eventually) detects the appearance of the desired configuration. The precise causal lineage is reconstructed using backtracking methods once the specified event has occurred in the system.

The idea behind a story is to retain only those events in the causal lineage that contributed a net progression towards the event of interest. This means in particular that circular histories are detected and eliminated. Circular histories generate a situation that is subsequently undone without leaving a side effect that impacts the causal lineage later on.

In sum, every good story has a beginning, a middle, and an end. A story is a sequence of events that:
- begins with the initial condition and ends with the observable event,
- consists only of events that are in the causal lineage to the observable (which eliminates events that are concurrent to the observable)
- contains no event sequence that revisits a situation encountered previously (which eliminates circles).

For the sake of simplicity, we now change the initial condition to consist of one instance of each agent. This yields two stories depending on whether K hits x or y first (Fig. 3 shows the former). If the initial condition contained more than one K, there would be a third story in which both sites are phosphorylated by different K-agents. Stories proceed in logical time, not physical time, and are therefore partial orders. The technical definition of causality between two events, $e_1 \rightarrow e_2$, means that (1) either e_2 could not have happened before e_1 or (2) e_2 would have prevented e_1 from happening. This corresponds to the intuition of one event (e_1) pointing "downstream" to another (e_2).[1] The partial nature of logical time seems well-suited for signalling networks in which a huge number of events do happen *concurrently*, making it difficult to spot the "organization" of events as implied by a set of mechanisms expressed by rules. It is intuitively clear that the very notion of signal has to do with the fact that some events enable or prevent others and therefore don't permute.

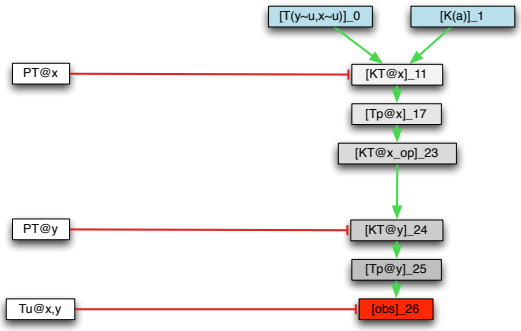


Figure 3: *A story: inhibitions or spoilers on the story are shown on the left. Event numbers represent the actual step in the simulation where these events occurred; missing events are either concurrent, or not represented because of successive opposite rules applications.*

2.5 Rule inhibition and activation maps

As seen with stories, drawing on the classical notion of causation, one obtains an *event-centric* view of a system. A more detailed view results from adding the classical concept of *conflict* between

[1] To keep things visually manageable, one represents only a reduced version of such relationships. The reduced set is such that its transitive closure regenerates the full orginal set of causal relations.

events. Two events are in conflict, if either event can happen, but not both. An example is given by the application of the rules `KT@x` and `PT@x`. Both rules potentially apply at the same time to a free instance of site x, but only one can actually occur. It is important to emphasize that conflict is defined over events (application instances of rules), not rules. Yet, the notion of conflict projects to rules in the sense that two rules conflict, if they can lead to conflicting events. Another way of viewing conflict at the rule level is to think of one rule inhibiting another rule, because application of the former destroys instances to which the latter could apply. The notion of conflict (or inhibition) between rules is not symmetric: rule X may conflict with rule Y, but not the other way around. The reason lies in aspects of a pattern on the left hand side of a rule that are only tested but not modified by it. The application of rule X may therefore alter aspects of a pattern that are only tested by rule Y. This is sufficient for X to destroy instances eligible for rule Y. Yet, the application of Y to those same instances would not make them ineligible for rule X.

Complementary to the notion of conflict is the notion of *activation*. Here, rule X is said to activate rule Y if an application of X *may* increase the number of instances to which rule Y applies. This means in particular that rule X changes aspects of a pattern that could, in some instance, have been the last ones missing for rule Y to apply. Yet, it may equally be the case that, in a given instance, rule X alone is not sufficient to make rule Y apply, but rule X contributes to moving the pattern closer to the specification given on the left hand side of rule Y. The *influence map* given in Fig. 4 recapitulates the main conflict and activation relationships between rules in the GK set.

2.6 Spoilers of stories

Because a story reflects causation between events, it lends itself as a scaffold for superimposing rules that are in conflict with it. As can be seen in Fig. 3, it is possible to represent *spoilers* of a given story by merging conflict relations from the inhibition map with a story of interest. These spoilers can actually pinpoint kinetic opportunities for delaying a story's ending or reducing its significance. For instance, as shown in Fig. 2(b), a tenfold increase in the rate constants of `PT@x` and `Tu@x` yields a much lower value for the observable.

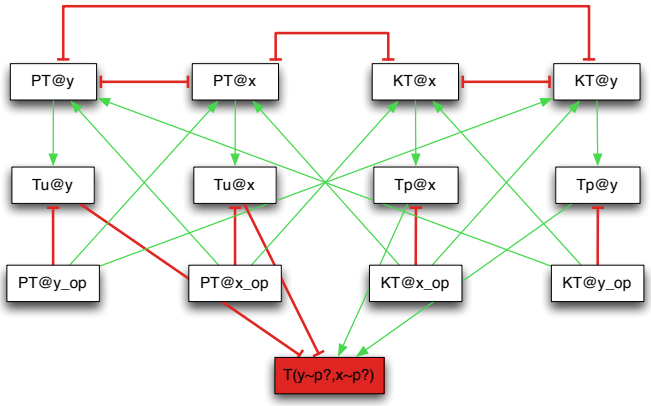


Figure 4: *The rule influence map of the Goldbeter-Koshland loop. Nodes are rules, edges represent causation (green), or conflict (red, not always symmetric). Trivial causations between a rule and its opposite are not shown. The final doubly phosphorylated state is represented at the bottom (red).*

This concludes the introduction of the main concepts underlying our rule-based approach to the analysis of signalling processes. We next proceed to apply these ideas to the more complex example of an EGF receptor model coupled to a MAP kinase cascade. In its full form, as presented in Ref. [16], EGFR signaling is a highly complex suite of pathways whose boundaries to other signaling systems appear increasingly blurred. While the present model falls far short of representing this complexity, it goes some way towards demonstrating how to eventually represent and face it.

3 EGFR model construction

The pathway to the activation of ERK has been the target of intense modelling efforts over the past decade. The ubiquity and importance of the pathway for biomedical applications [20, Chap. 5] have spurred extensive studies at the mechanistic level of protein-protein interactions and localizations. At the same time, the subtleties uncovered by these investigations (see, for example, the receptor network combinatorics of the pathway [13]) have made it clear that intuition alone, however sharp, cannot front the complexity and only risks flushing enormous amounts of drug development money down the drain. To calibrate the reader on the magnitudes involved, the particular model presented below generates about $400,000$ different (non-isomorphic) molecular species. An exhaustive approach with differential equations would, in principle, require that many equations. And that is in no way a large example.

In outline, a signal arrives at the cell membrane in the form of a ligand, EGF, which binds to the extra-cellular portion of a special receptor protein, EGFR, that straddles the membrane. With the arrival of EGF, an EGFR becomes capable of binding to a neighbouring EGFR also bound to a ligand. Such receptor pairs can cross-activate one another, meaning that certain of their intracellular residues become phosphorylated. These phosphorylated residues now serve as binding sites for a variety of proteins in the cytoplasm. This in turn leads to the activation of a small protein, Ras, that serves as a kind of relay for triggering a cascade of phosphorylations which comprises three stacked GK loops in which the fully phosphorylated form of one loop acts as the kinase of the subsequent loop, causing the overall cascade to behave like an amplifier and culminating in the activation of ERK.

Fig. 5 depicts the contact map of a substantial body of rules describing also the AKT pathway downstream the same family of receptors. The associated rule-based model overlays a causal structure on the contact map by specifying, for example, that Ras can only bind Raf if it has been phosphorylated beforehand at site S1S2. For the purpose of the present paper, we will restrict ourselves to the subset of rules relevant to the two left components of the contact map (Fig. 5). The rule set itself was mainly obtained from refactoring two existing ordinary differential equations (ODE) models [3, 19] treating two different aspects of EGF-EGFR signalling: down-regulation of Erk activity through a negative feedback and down-regulation of the signal through internalization. Refactoring involves searching the literature for mechanistic details germane to the interactions represented in ODE models. It was surprisingly easy to combine both components into one κ-model.

In its present form, our basic EGFR scheme consists of a:
- receptor module, which for illustration purposes retains only the EGFR (or ErbB1) receptor, but includes internalization dynamics through (at the moment) fictitious "sites" whose state flags localization. This module can be refined to include a receptor heterodimerization network comprising all four receptors of the ErbB family.
- adaptors and relays, such as Sos, Grb2, and Ras. These too can be extended as the complexity of the model is built up in a stepwise fashion.

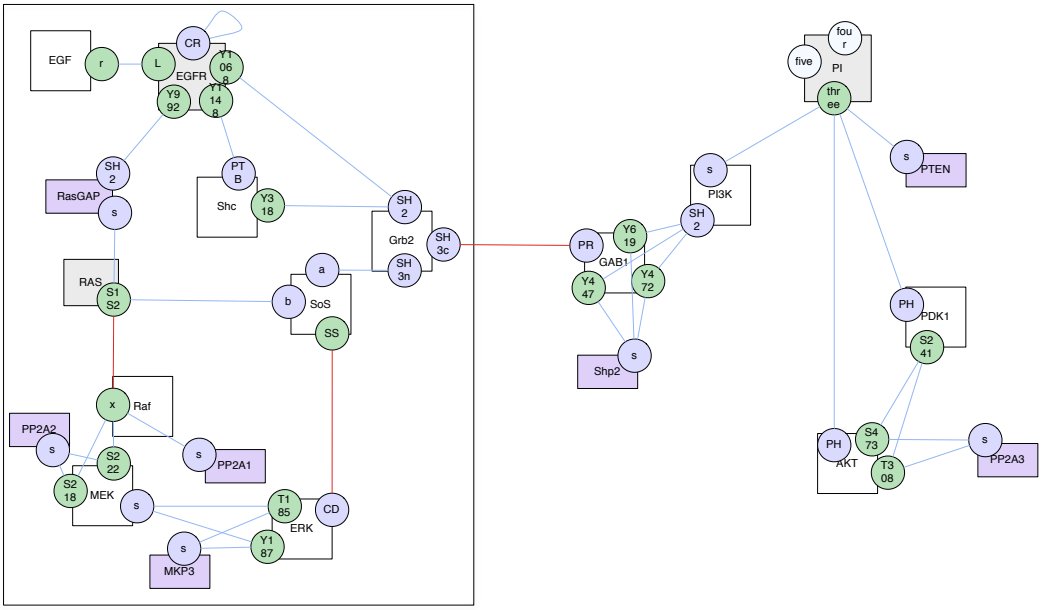


Figure 5: *The contact map of the joint AKT and ERK pathways: edges represent potential bindings between sites; nodes are the agents of the model; sites can be modified, or bound (blue), or both (green). In this paper we are only concerned with the left components and their interaction via the negative feedback arising from the CD-SS edge from ERK to SoS (in red). The rule set is given in the appendix.*

- the target MAPK cascade.

Our basic EGFR model contains 68 rules (see appendix for the complete rule set), which generate more than $400,000$ distinct complexes.

4 Causality

4.1 Ras activation

An examination of the contact map immediately reveals a potentially critical role for Ras in the behaviour of the model: Ras has only one site but can bind to three distinct agents (SoS, RasGAP and Raf) and so probably forms a bottleneck. Inspection of the rules governing these four agents allows us to refine the contact map: Ras's site has an internal state (representing active or inactive), only SoS can bind to an inactive Ras, whereas RasGAP and Raf must compete for active Ras.

In terms of signal propagation, SoS activates Ras so that Ras can in turn activate Raf. RasGAP plays an inhibitory role by deactivating Ras. We can observe this chain of causality by extracting stories for rules expressing Ras's recruitment of Raf (Fig. 6 and 7).

RasGAP does not appear in either of these stories, confirming that it plays no role in the logical propagation of the signal. RasGAP, however, does play a role in shaping the kinetics of signal propagation. Indeed, most sequences of events leading to Raf's recruitment do exhibit RasGAP

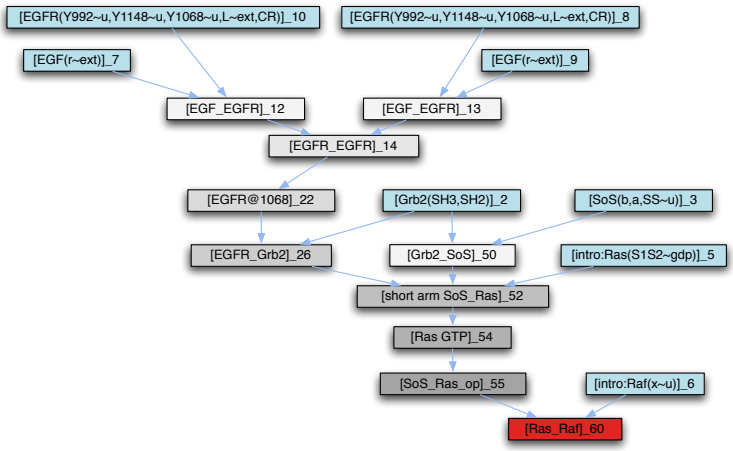


Figure 6: *Short arm activation of Ras (without Shc)*

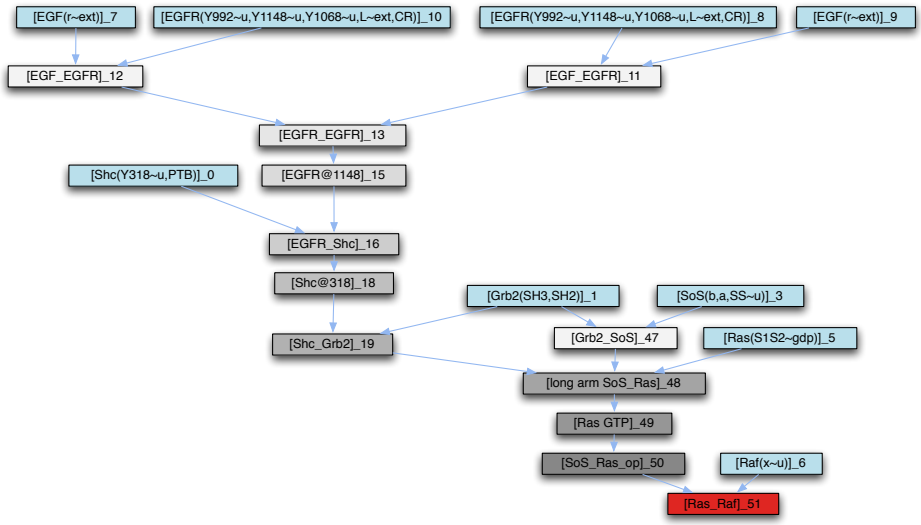


Figure 7: *Long arm activation of Ras (with Shc)*

intervention, a typical example being given Fig. 8. This example emphasizes a slightly paradoxical effect of the EGF signal: in order to propagate the signal, SoS is induced to activate Ras (and hence the downstream cascade to ERK) but, at the same time, the signal also induces RasGAP to frustrate SoS's work. Of course, a signal needs to be controlled and eventually down-regulated, so the existence of a RasGAP-like agent should be expected. Yet, the fact that both the positive (SoS) and the negative (RasGAP) influences on Ras depend on the same signal suggests that Ras's activation dynamics only depend on the relative concentrations of SoS and RasGAP: if RasGAP dominates, Ras activation will be weak and short-lived; if SoS dominates, it will be stronger and

last longer.

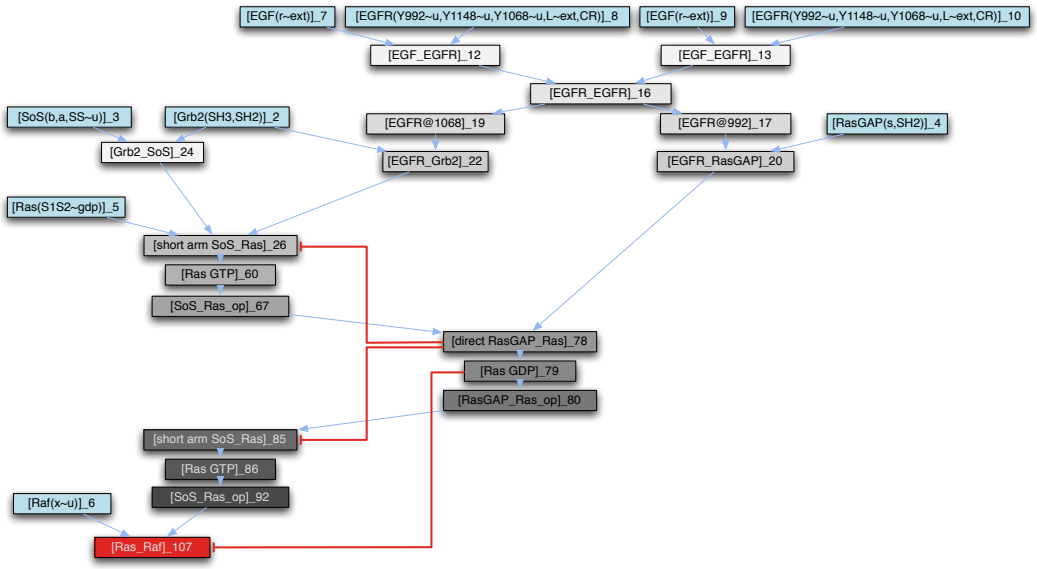


Figure 8: *Battle between SoS and RasGAP*

4.2 SoS deactivation

SoS has a second enemy in the form of activated ERK, the endpoint of this pathway that SoS has contributed to. Activated ERK phosphorylates SoS, inhibiting the formation of the complex between Grb2 and SoS (Fig. 9).

For the duration of the SoS's phosphorylation, this substantially weakens the signal from SoS to Ras, essentially shifting the balance in favour of RasGAP. As a result, the level of active Ras decreases which, with a small delay, causes a significant reduction in active ERK at the bottom of the cascade. At that moment, SoS is no longer being strongly targeted by ERK and can, once more, signal to Ras. We thus see a cyclic process of activation and then inhibition of Ras, leading to an oscillating activation pattern for ERK, typical of a cascade embedded in a negative feedback loop [14].

The crucial parameter determining the shape of these oscillations is the "recovery rate" of SoS from its phosphorylation by ERK. A slow recovery rate leads to a clear oscillation of the cascade. As the rate of recovery increases, the cascade oscillates more quickly, albeit with the same amplitude. With a sufficiently rapid recovery rate, the cascades achieves a transient activation, again with the same amplitude, with a little oscillation as the signal dies (Fig. 10).

4.3 ERK and MKP3

Another factor regulating the effect of the negative feedback to SoS comes from MKP3, the phosphatase targeting ERK, as suggested by the spoilers shown in Fig. 9. A higher concentration of

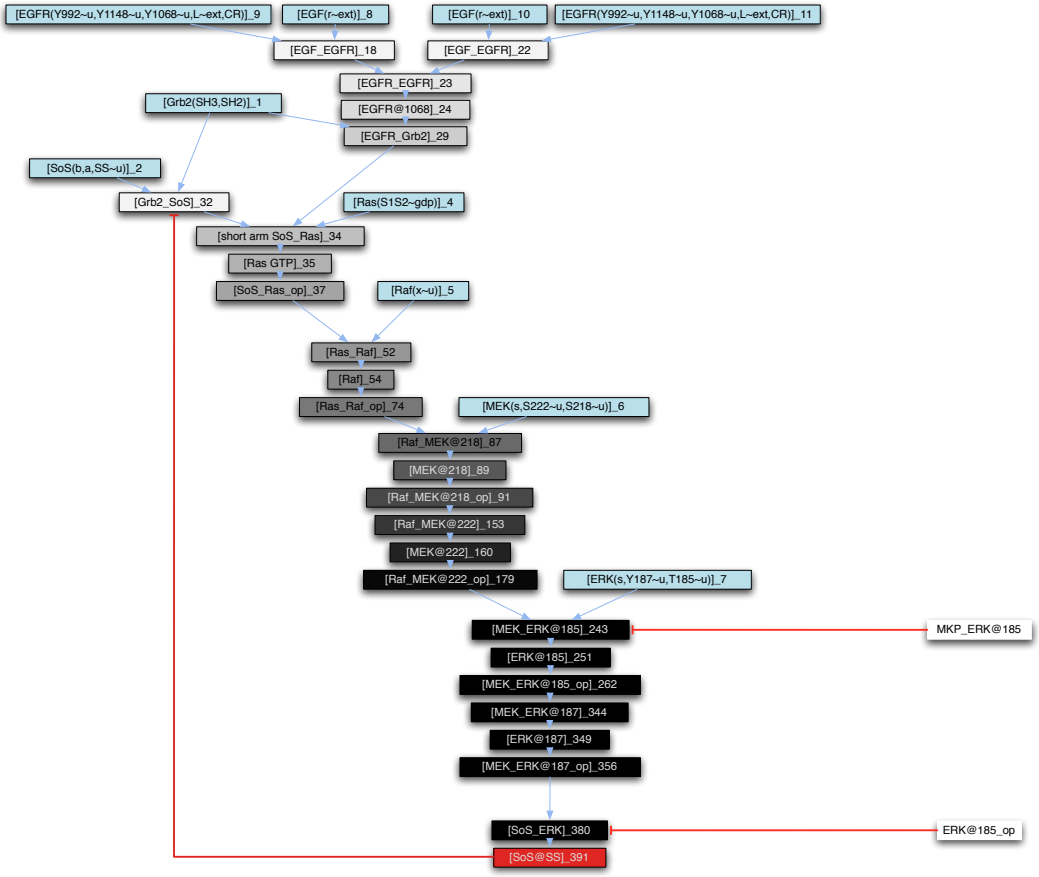
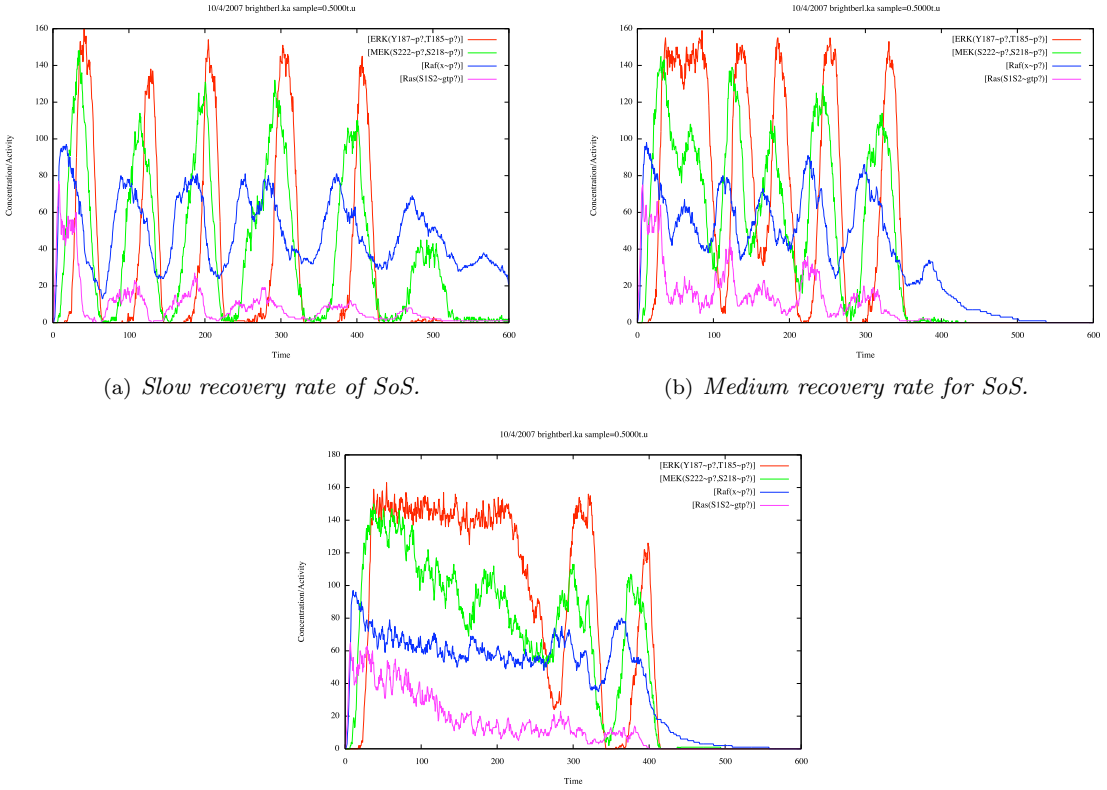


Figure 9: *Negative feedback from active ERK to SoS*

MKP3 will tend to hamper activation of ERK and this impacts the rate of oscillation at the cascade: increasing the concentration of MKP3 over successive simulations, one observes (without surprise) that the amplitude of ERK activation decreases. However, one also observes that ERK remains active for less time, leading to a gradual increase in the frequency of oscillation (Fig. 11). In addition, the signal attenuates faster: with more phosphatase in the system, ERK requires a higher "threshold" concentration of its kinase (MEK) in order to achieve significant activation.

While this obviously constitutes a spoiler of ERK activation, it also protects SoS from being in turn spoiled by ERK. However, this secondary effect turns out to be fairly minor, since a typical system contains more ERK than SoS molecules. Therefore, even a reduced level of ERK activation suffices to mount a powerful spoiling of SoS.

One final parameter substantially influences ERK's level of activation: the speed of SoS phosphorylation by ERK. A slow rate limits the effect of the negative feedback, leading to a longer and steadier period of Ras and ERK activation and little to no oscillation. A fast rate accentuates the negative feedback effect, considerably shortening the time during which Ras signals and, in tandem



(a) *Slow recovery rate of SoS.*

(b) *Medium recovery rate for SoS.*

(c) *Fast recovery rate for SoS.*

Figure 10: Oscillations and rates of SoS recovery.

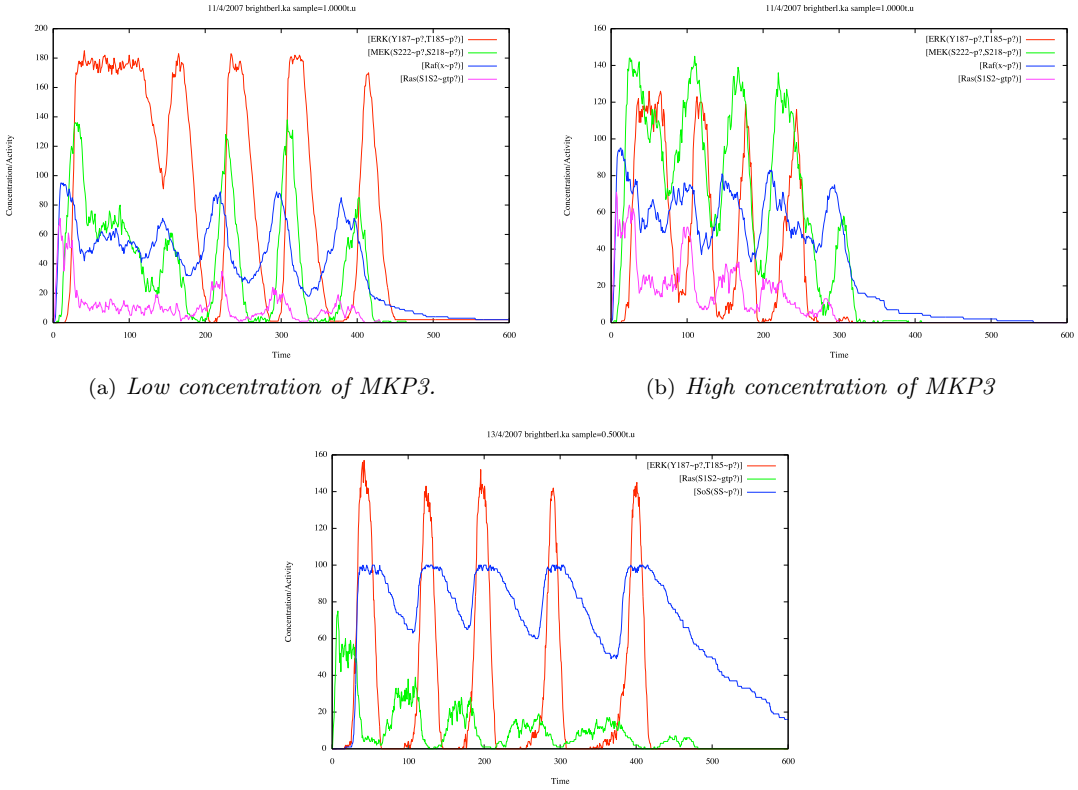
with a slow recovery rate, leads to a pronounced, low-frequency oscillation (Fig. 11).

5 Conclusions

We have illustrated how rule-based modeling transcends the bottleneck of the traditional ODE-based framework. It does so in many important ways both practical and conceptual, which we summarize here.

Rule-based modeling tames the combinatorial explosion by decontextualizing reactions between molecular species into rules defined on patterns. As an immediate corollary, modifications and extensions become fairly straightforward.

Rules represent nuggets of mechanistic knowledge that current experimental practice is rapidly accumulating. Rather than expressing such knowledge in terms of human language or non-actionable graphical information, it seems vastly more useful to represent it in a context-free grammar ready for computational consumption. Much like chemical reactions, rules can be viewed as operational "instructions" that can be let loose on a set of initial molecular agents, driving the unfolding of pathways and their kinetics. In this sense, κ-rules make knowledge executable. The granularity of



(a) *Low concentration of MKP3.*

(b) *High concentration of MKP3*

(c) *Combination of fast feedback and slow recovery*

Figure 11: Impact of MKP3 concentration on pathway oscillations.

such rules is, in principle, adaptable to the needs of bench scientists. We believe that the current level of granularity offered by κ or the variant langugae BNG [2] meets the most urgent practical needs.

Sets of rules are not just inputs to simulators, like systems of differential equations are not just inputs to numerical integrators. Rather, rule sets replace systems of differential equations as formal entities that can be subject to rigorous analysis from which to extract predictive and explanatory information about the behavior of systems. In contrast to the synchronicity of differential equations, rules operate in a concurrent execution model, which is a far more appropriate representation of what is actually going on in cells. This constitutes rather unfamiliar terrain for many biologists. Yet, this is the turf that has been successfully plowed for over thirty years in computer science. We have shown how notions of conflict and causation can be used to build maps that relate rules with one another. We have defined a concept of story, which we believe formalizes the intuitive notion of "pathway" that biologists entertain. Stories are partial orders representing causal lineages that explain how a given observable arises in logical time. Stories change in time, since they depend on available molecular resources. The superposition of stories with rule inhibition maps identifies potential "story spoilers", junctures at which logical structure meets kinetics. We have

made extensive use of this trick to explain several dynamical and logical features of a simplified EGFR signalling system. Our experience is that this mode of reasoning matches quite naturally the way biologists intuitively go about telling their "stories". The only difference is that our framework formalizes this process and therefore enables computational procedures to tackle much more complicated systems in rigorous ways when the story-telling of biologists risks degenerating into just that.

It is worth emphasizing that the main dynamic characteristics of our modest EGFR case were obtained with uniform rate constants for all rules. The impact of certain rules on these characteristics was then explored by dialing up different rate constants. This is not to say that rate constants don't matter, but it does hint at the importance of the causal architecture of a system in shaping dynamics. Disentangling the contribution of causal structure and rates to overall systems dynamics is hardly possible in large systems of differential equations. By forgoing this separation, modelers who fit rate constants to ODE systems risk to engage in an idle encoding exercise rather than a modeling process, since many behaviors can be inscribed into any sufficiently large system of ODEs by appropriate choice of rate parameters. We believe that rule-based modeling affords a strategy whereby one first tries to get the logical structure to generate key dynamical characteristics and then tunes the rate constants to obtain the fine structure. If the logical structure is insufficient, odds are that our knowledge is insufficient and that more experiments are better than more models.

It is useful to think of rule-based modeling as a task in concurrent programming, in which rules are computational instructions that contribute to system behavior. It is difficult to grasp how concurrent systems – in particular natural ones like cells, tissues, and organisms – function and why they function the way they do. Modeling in a rule-based format yields a better appreciation of the role played by individual mechanisms in generating collective behavior. Linking architecture to behavior will produce more informed strategies for intervention in the case of disease and will help us distill the principles that enabled cells to evolve such versatile information processing systems in the first place. Like in programming, however, there is ample opportunity for mistakes. In fact, a model might be wrong not because it isn't a correct description of the world, but because it may not express what the modeler intended (think typo). To catch such mistakes is crucial and will eventually necessitate a veritable "modeling environment" with sophisticated debugging and verification tools. The complete absence of such tools in the traditional ODE framework, makes classic models beyond a certain size highly prone to error and exceedingly difficult to maintain. As in programming, rule-based models are "grown" by merging smaller models to build larger models that are easily refined by incorporating new empirical knowledge. Rule-based modeling is as much a scientific instrument as it is effective knowledge management.

References

[1] C. Baldi, Pierpaolo Degano, and Corrado Priami. Causal pi-calculus for biochemical modeling. In *Proceedings of the AI*IA Workshop on BioInformatics 2002*, pages 69–72, 2002.

[2] M.L. Blinov, J.R. Faeder, and W.S. Hlavacek. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20:3289–3292, 2004.

[3] F.A. Brightman and D.A. Fell. Differential feedback regulation of the MAPK cascade underlies the quantitative differences in EGF and NGF signalling in PC12 cells. *FEBS Lett*, 482(3):169–174, 2000.

[4] Pierre-Louis Curien, Vincent Danos, Jean Krivine, and Min Zhang. Computational self-assembly. Submitted, Feb 2007.

[5] Vincent Danos, Jérôme Feret, Walter Fontana, and Jean Krivine. Scalable modelling of biological pathways. Submitted, Feb 2007.

[6] Vincent Danos, Jérôme Feret, Walter Fontana, and Jean Krivine. What is so special about biological signalling? Submitted, Apr 2007.

[7] Vincent Danos and Cosimo Laneve. Core formal molecular biology. In *Proceedings of the 12th European Symposium on Programming, ESOP'03*, volume 2618 of *LNCS*, pages 302–318. Springer-Verlag, April 2003.

[8] Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, September 2004.

[9] P. Degano and C. Priami. Causality for mobile processes. *Proceedings of ICALP*, 95:660–671, 1995.

[10] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and K. Sonmez. Pathway logic: Symbolic analysis of biological signaling. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 400–412, January 2002.

[11] A. Goldbeter and D.E. Koshland. An Amplified Sensitivity Arising from Covalent Modification in Biological Systems. *Proceedings of the National Academy of Sciences*, 78(11):6840–6844, 1981.

[12] C.Y.F. Huang and J.E. Ferrell. Ultrasensitivity in the mitogen-activated protein kinase cascade, 1996.

[13] N.E. Hynes and H.A. Lane. ERBB receptors and cancer: the complexity of targeted inhibitors. *Nature Reviews Cancer*, 5(5):341–354, 2005.

[14] B.N. Kholodenko. Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades, 2000.

[15] B.N. Kholodenko, O.V. Demin, G. Moehren, and J.B. Hoek. Quantification of Short Term Signaling by the Epidermal Growth Factor Receptor. *Journal of Biological Chemistry*, 274(42):30169–30181, 1999.

[16] Kanae Oda, Yukiko Matsuoka, Akira Funahashi, and Hiroaki Kitano. A comprehensive pathway map of epidermal growth factor receptor signaling. *Molecular Systems Biology*, 1, May 2005.

[17] Richard J. Orton, Oliver E. Sturm, Vladislav Vyshemirsky, Muffy Calder, David R. Gilbert, and Walter Kolch. Computational modelling of the receptor tyrosine kinase activated MAPK pathway. *Biochemical Journal*, 392(2):249–261, 2005.

[18] T. Pawson and P. Nash. Assembly of Cell Regulatory Systems Through Protein Interaction Domains. *Science*, 300(5618):445–452, 2003.

[19] Birgit Schoeberl, Claudia Eichler-Jonsson, Ernst-Dieter Gilles, and Gertraud Müller. Computational modeling of the dynamics of the map kinase cascade activated by surface and internalized EGF receptors. *Nature Biotechnology*, 20:370–375, 2002.

[20] Robert A. Weinberg. *The Biology of Cancer*. Garland Science, Jun 2006.

[21] Glyn Winskel. An introduction to event structures. In *REX Workshop 1988*, LNCS. Springer Verlag, 1989.

6 Appendix: the model code

This appendix contains the κ rendition of the Schoeberl et al. (2002) EGFR model [19], combined with the negative feedback of the earlier Brightman & Fell (2000) model [3]. Rule names used in the body and figures of the paper are defined below. Rates were all set to 1, unless specified differently in the rule set or in the nunerical experiments reported in the paper.

Some rules below use an additional `!_` notation, meaning 'bound to something', as a convenient way to shorten rules.

```
# phase -> activated dimers
# external dimers:
'EGF_EGFR'  EGF(r~ext), EGFR(L~ext,CR) <-> EGF(r~ext!1), EGFR(L~ext!1,CR)
'EGFR_EGFR' EGFR(L~ext!_,CR), EGFR(L~ext!_,CR) <-> EGFR(L~ext!_,CR!1), EGFR(L~ext!_,CR!1)
# simplified phosphorylation (internal or external)
'EGFR@992'  EGFR(CR!_,Y992~u) -> EGFR(CR!_,Y992~p)
'EGFR@1068' EGFR(CR!_,Y1068~u) -> EGFR(CR!_,Y1068~p)
'EGFR@1148' EGFR(CR!_,Y1148~u) -> EGFR(CR!_,Y1148~p)
# simplified dephosphorylation (internal or external)
'992_op'    EGFR(Y992~p) -> EGFR(Y992~u)
'1068_op'   EGFR(Y1068~p) -> EGFR(Y1068~u)
'1148_op'   EGFR(Y1148~p) -> EGFR(Y1148~u)

# phase -> internalization, degradation and recycling
# internalization:
'int_monomer'  EGF(r~ext!1), EGFR(L~ext!1,CR) -> EGF(r~int!1), EGFR(L~int!1,CR) @ 0.02
'int_dimer'    EGF(r~ext!1), EGFR(L~ext!1,CR!2), EGF(r~ext!3), EGFR(L~ext!3,CR!2) -> \
               EGF(r~int!1), EGFR(L~int!1,CR!2), EGF(r~int!3), EGFR(L~int!3,CR!2) @ 0.02
# dissociation:
'EGFR_EGFR_op' EGFR(L~int!_,CR!1), EGFR(L~int!_,CR!1) -> EGFR(L~int!_,CR), EGFR(L~int!_,CR)
'EGF_EGFR_op'  EGF(r~int!1), EGFR(L~int!1,CR) -> EGF(r~int), EGFR(L~int,CR)
# degradation:
'deg_EGF'      EGF(r~int) ->
'deg_EGFR'     EGFR(L~int,CR) ->
# recycling:
'rec_EGFR'     EGFR(L~int,Y992~u,Y1068~u,Y1148~u) -> EGFR(L~ext,Y992~u,Y1068~u,Y1148~u)

# phase -> SoS and RasGAP recruitment
'EGFR_RasGAP'  EGFR(Y992~p), RasGAP(SH2) <-> EGFR(Y992~p!1), RasGAP(SH2!1)
'EGFR_Grb2'    EGFR(Y1068~p), Grb2(SH2) <-> EGFR(Y1068~p!1), Grb2(SH2!1)
'Grb2_SoS'     Grb2(SH3), SoS(a,SS~u) -> Grb2(SH3!1), SoS(a!1,SS~u)
'Grb2_SoS_op'  Grb2(SH3!1), SoS(a!1) -> Grb2(SH3), SoS(a)
'EGFR_Shc'     EGFR(Y1148~p), Shc(PTB) <-> EGFR(Y1148~p!1), Shc(PTB!1)
'Shc_Grb2'     Shc(Y318~p), Grb2(SH2) <-> Shc(Y318~p!1), Grb2(SH2!1)
'Shc@318'      EGFR(CR!_,Y1148~p!1), Shc(PTB!1,Y318~u) -> EGFR(CR!_,Y1148~p!1), Shc(PTB!1,Y318~p)
'Shc@318_op'   Shc(Y318~p) -> Shc(Y318~u)

# phase -> active Ras
# activate:
'long arm SoS_Ras'  EGFR(Y1148~p!1), Shc(PTB!1,Y318~p!2), Grb2(SH2!2,SH3!3), SoS(a!3,b), Ras(S1S2~gdp) -> \
                    EGFR(Y1148~p!1), Shc(PTB!1,Y318~p!2), Grb2(SH2!2,SH3!3), SoS(a!3,b!4), Ras(S1S2~gdp!4)
'short arm SoS_Ras' EGFR(Y1068~p!1), Grb2(SH2!1,SH3!2), SoS(a!2,b), Ras(S1S2~gdp) -> \
                    EGFR(Y1068~p!1), Grb2(SH2!1,SH3!2), SoS(a!2,b!3), Ras(S1S2~gdp!3)
'Ras GTP'           SoS(b!1), Ras(S1S2~gdp!1) -> SoS(b!1), Ras(S1S2~gtp!1)
'SoS_Ras_op'        SoS(b!1), Ras(S1S2!1) -> SoS(b), Ras(S1S2)

# deactivate:
'direct RasGAP_Ras' EGFR(Y992~p!1), RasGAP(SH2!1,s), Ras(S1S2~gtp) -> \
                    EGFR(Y992~p!1), RasGAP(SH2!1,s!2), Ras(S1S2~gtp!2)
'Ras GDP'           RasGAP(s!1), Ras(S1S2~gtp!1) -> RasGAP(s!1), Ras(S1S2~gdp!1)
```

```
'RasGAP_Ras_op'     RasGAP(s!1), Ras(S1S2!1) -> RasGAP(s), Ras(S1S2)
'intrinsic Ras GDP' Ras(S1S2~gtp) -> Ras(S1S2~gdp)

# phase -> active Raf
# activation:
'Ras_Raf'      Ras(S1S2~gtp), Raf(x~u) -> Ras(S1S2~gtp!1), Raf(x~u!1)
'Raf'          Ras(S1S2~gtp!1), Raf(x~u!1) -> Ras(S1S2~gtp!1), Raf(x~p!1)
'Ras_Raf_op'   Ras(S1S2~gtp!1), Raf(x!1) -> Ras(S1S2~gtp), Raf(x)
# deactivation:
'PP2A1_Raf'    PP2A1(s), Raf(x~p) -> PP2A1(s!1), Raf(x~p!1)
'Raf_op'       PP2A1(s!1), Raf(x~p!1) -> PP2A1(s!1), Raf(x~u!1)
'PP2A1_Raf_op' PP2A1(s!1), Raf(x!1) -> PP2A1(s), Raf(x)

# phase -> active MEK
# activation:
'Raf_MEK@222'      Raf(x~p), MEK(S222~u) -> Raf(x~p!1), MEK(S222~u!1)
'MEK@222'          Raf(x~p!1), MEK(S222~u!1) -> Raf(x~p!1), MEK(S222~p!1)
'Raf_MEK@222_op'   Raf(x~p!1), MEK(S222!1) -> Raf(x~p), MEK(S222)
'Raf_MEK@218'      Raf(x~p), MEK(S218~u) -> Raf(x~p!1), MEK(S218~u!1)
'MEK@218'          Raf(x~p!1), MEK(S218~u!1) -> Raf(x~p!1), MEK(S218~p!1)
'Raf_MEK@218_op'   Raf(x~p!1), MEK(S218!1) -> Raf(x~p), MEK(S218)
# deactivation:
'PP2A2_MEK@222'    PP2A2(s), MEK(S222~p) -> PP2A2(s!1), MEK(S222~p!1)
'MEK@222_op'       PP2A2(s!1), MEK(S222~p!1) -> PP2A2(s!1), MEK(S222~u!1)
'PP2A2_MEK@222_op' PP2A2(s!1), MEK(S222!1) -> PP2A2(s), MEK(S222)
'PP2A2_MEK@218'    PP2A2(s), MEK(S218~p) -> PP2A2(s!1), MEK(S218~p!1)
'MEK@218_op'       PP2A2(s!1), MEK(S218~p!1) -> PP2A2(s!1), MEK(S218~u!1)
'PP2A2_MEK@218_op' PP2A2(s!1), MEK(S218!1) -> PP2A2(s), MEK(S218)

# phase -> active ERK
# activation:
'MEK_ERK@185'    MEK(s,S218~p,S222~p), ERK(T185~u) -> MEK(s!1,S218~p,S222~p), ERK(T185~u!1)
'ERK@185'        MEK(s!1,S218~p,S222~p), ERK(T185~u!1) -> MEK(s!1,S218~p,S222~p), ERK(T185~p!1)
'MEK_ERK@185_op' MEK(s!1), ERK(T185!1) -> MEK(s), ERK(T185)
'MEK_ERK@187'    MEK(s,S218~p,S222~p), ERK(Y187~u) -> MEK(s!1,S218~p,S222~p), ERK(Y187~u!1)
'ERK@187'        MEK(s!1,S218~p,S222~p), ERK(Y187~u!1) -> MEK(s!1,S218~p,S222~p), ERK(Y187~p!1)
'MEK_ERK@187_op' MEK(s!1), ERK(Y187!1) -> MEK(s), ERK(Y187)
# deactivation:
'MKP_ERK@185'    MKP3(s), ERK(T185~p) -> MKP3(s!1), ERK(T185~p!1)
'ERK@185_op'     MKP3(s!1), ERK(T185~p!1) -> MKP3(s!1), ERK(T185~u!1)
'MKP_ERK@185_op' MKP3(s!1), ERK(T185!1) -> MKP3(s), ERK(T185)
'MKP_ERK@187'    MKP3(s), ERK(Y187~p) -> MKP3(s!1), ERK(Y187~p!1)
'ERK@187_op'     MKP3(s!1), ERK(Y187~p!1) -> MKP3(s!1), ERK(Y187~u!1)
'MKP_ERK@187_op' MKP3(s!1), ERK(Y187!1) -> MKP3(s), ERK(Y187)

# negative feedback
'SoS_ERK'    SoS(SS~u), ERK(s,T185~p,Y187~p) -> SoS(SS~u!1), ERK(s!1,T185~p,Y187~p)
'SoS_ERK_op' SoS(SS!1), ERK(s!1) -> SoS(SS), ERK(s)
# feedback creation
'SoS@SS'     SoS(SS~u!1), ERK(s!1,T185~p,Y187~p) -> SoS(SS~p!1), ERK(s!1,T185~p,Y187~p)
```

```
# feedback recovery
'SoS@SS_op'  SoS(SS~p) -> SoS(SS~u)
```

The initial state used in the various simulations (unless specified otherwise) is shown below. Techniques used for the simulation are explained in Ref. [5]. Stories were extracted by sampling trajectories using lower numbers of agents (as in the GK rule set case) and eliminating concurrent events, and cycles.

```
%init: 10*(EGF(r~ext)) + 100*(EGFR(L~ext,CR,Y992~u,Y1068~u,Y1148~u)) + 100*(Shc(PTB,Y318~u)) \
+ 100*(Grb2(SH2,SH3!1),SoS(a!1,b,SS~u)) + 200*(RasGAP(SH2,s)) + 100*(Ras(S1S2~gdp)) \
+ 100*(Raf(x~u)) + 25*(PP2A1(s)) + 50*(PP2A2(s)) \
+ 200*(MEK(s,S222~u,S218~u)) + 200*(ERK(s,T185~u,Y187~u)) + 50*(MKP3(s))
```